USING MACHINE LEARNINIG MODEL

Online Payments
Fraud Detection

Presentation By

WANAIM essaadia

Under the guidance of

Pr.KHamjane

Introduction

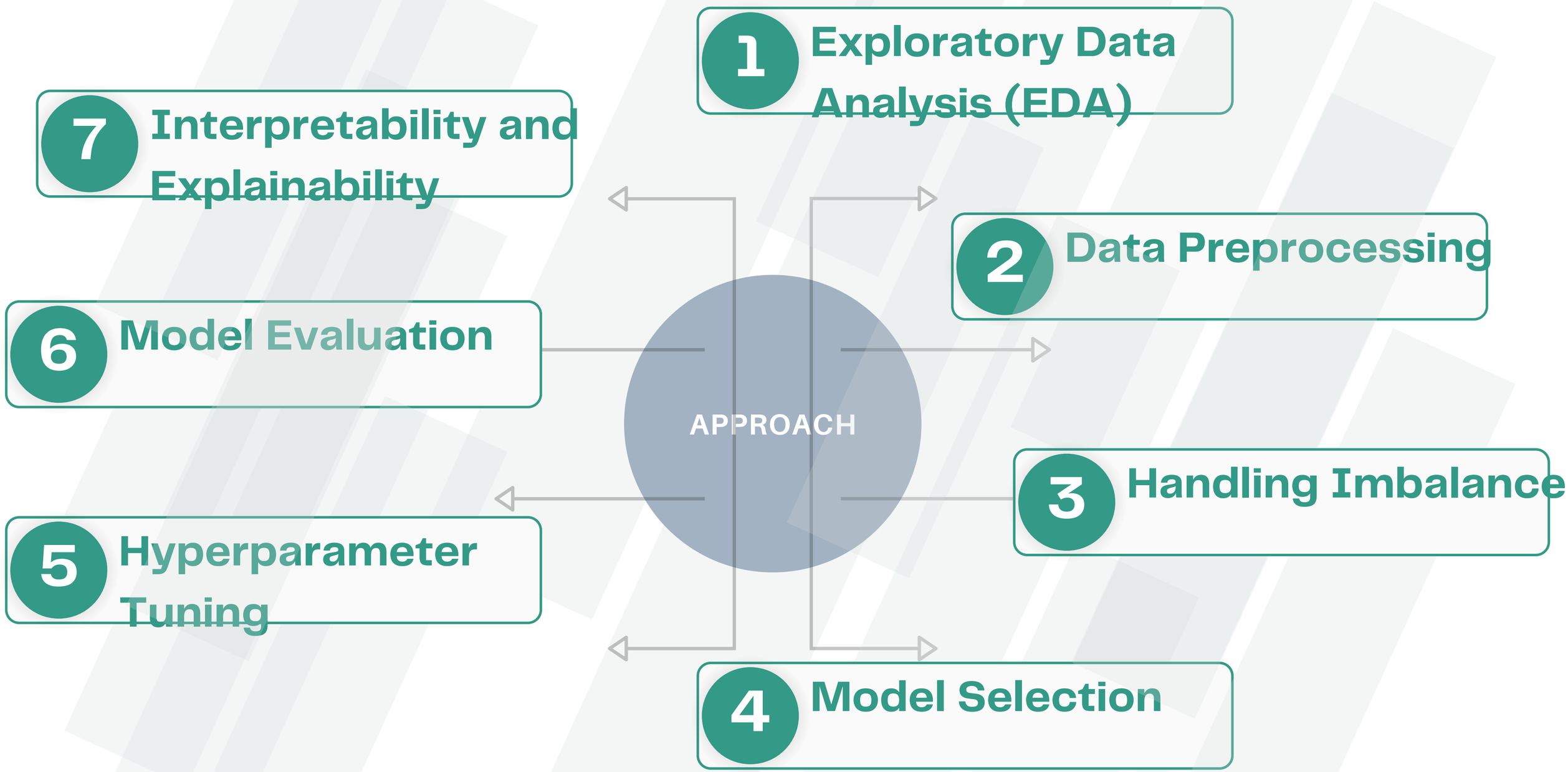
Online payment is the most popular transaction method in the world today. however, with an increase in online payments also comes a rise in payment fraud.

To identify online payment fraud we need to train a model to classify fraudulent & non fraudulent payment.

1 Data Description

Explanation of features:

- step: represents a unit of time where 1 step equals 1 hour.
- type: online transaction type
- amount: the amount of the transaction
- nameOrig: client starting transaction
- oldbalanceOrg: balance before transaction
- newbalanceOrig: balance after transaction
- nameDest: recipient of the transaction
- oldbalanceDest: initial balance of the recipient before the transaction
- newbalanceDest: the recipient's new balance after the transaction
- isFraud: fraudulent transaction

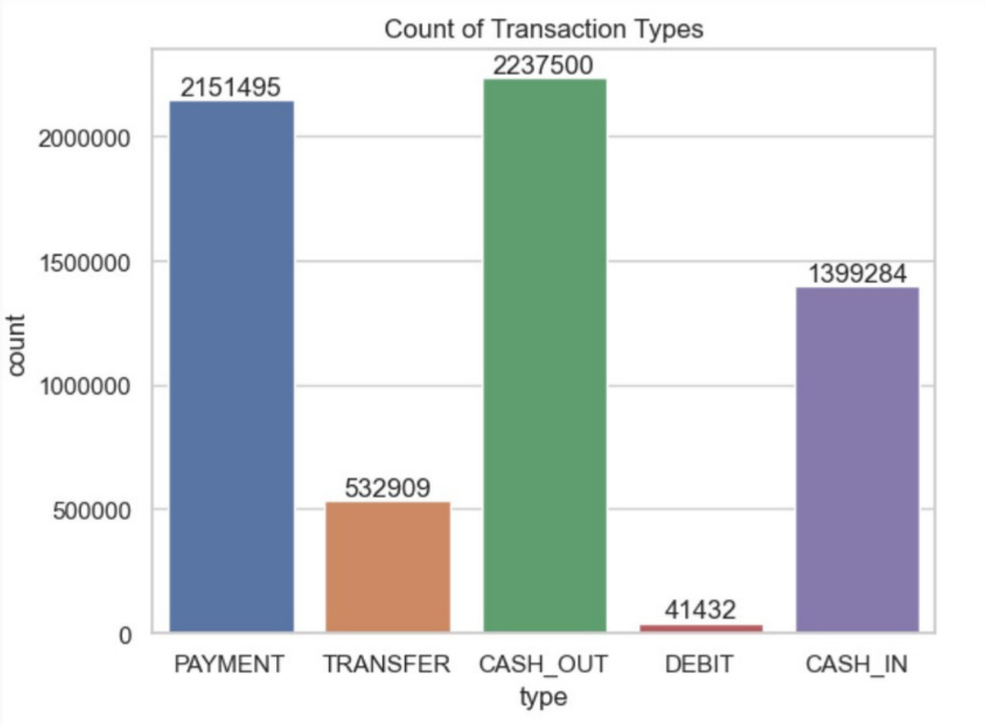

APPROACH
1 Exploratory Data Analysis (EDA)
2 Data Preprocessing
3 Handling Imbalance
4 Model Selection
5 Hyperparameter Tuning
6 Model Evaluation
7 Interpretability and Explainability

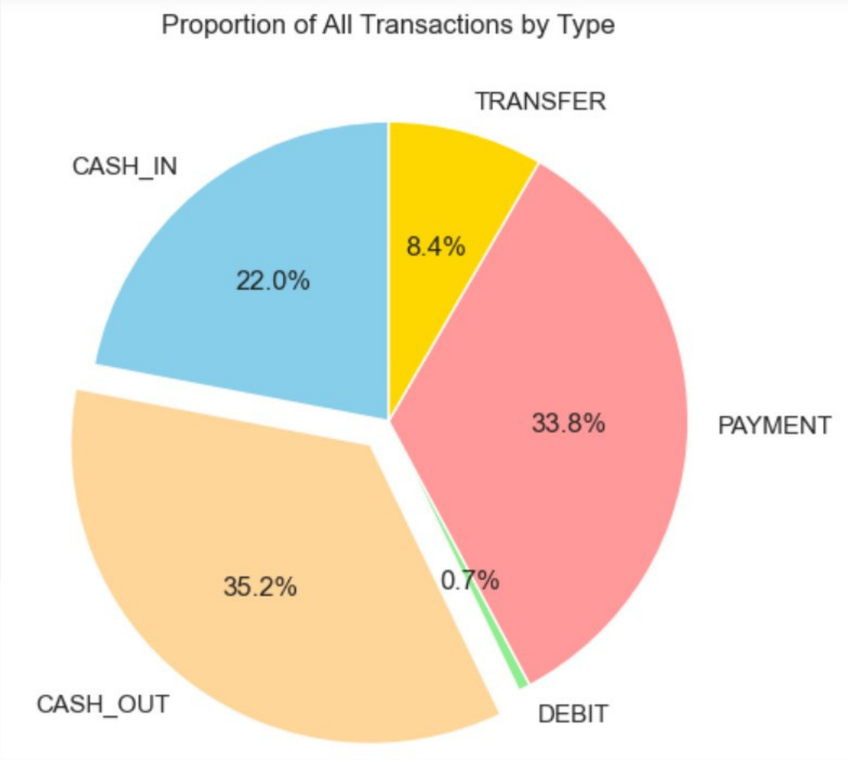
1 Exploratory Data Analysis

- The dataset consists of 6,362,620 observations.
- There are no missing values.
- There are no duplicate values

Exploratory Data Analysis

The chart illustrates the proportion of each transaction type relative to the total number of transactions.
There are five distinct transaction types in the dataset: PAYMENT, TRANSFER, CASH_OUT, DEBIT, and CASH_IN

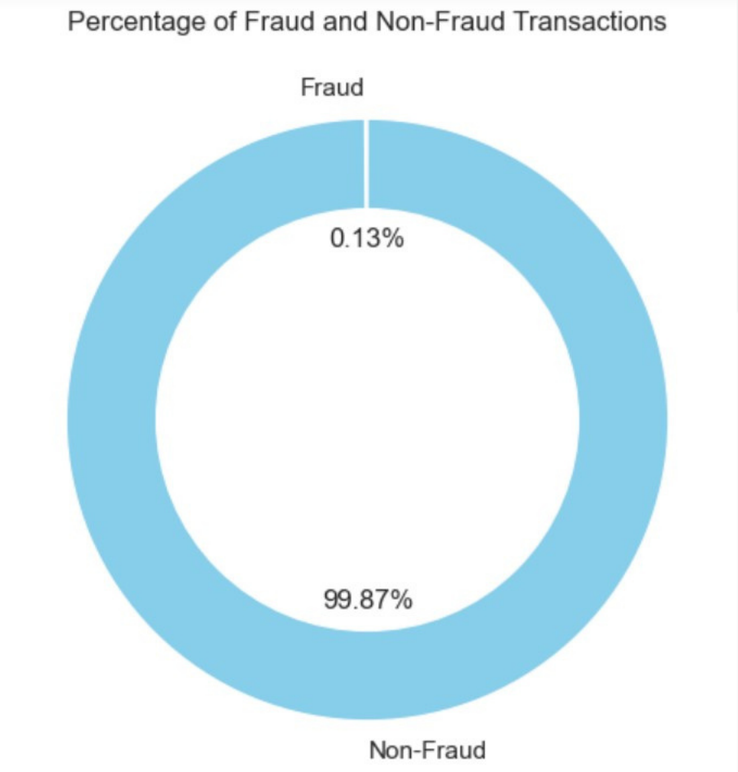




Exploratory Data Analysis

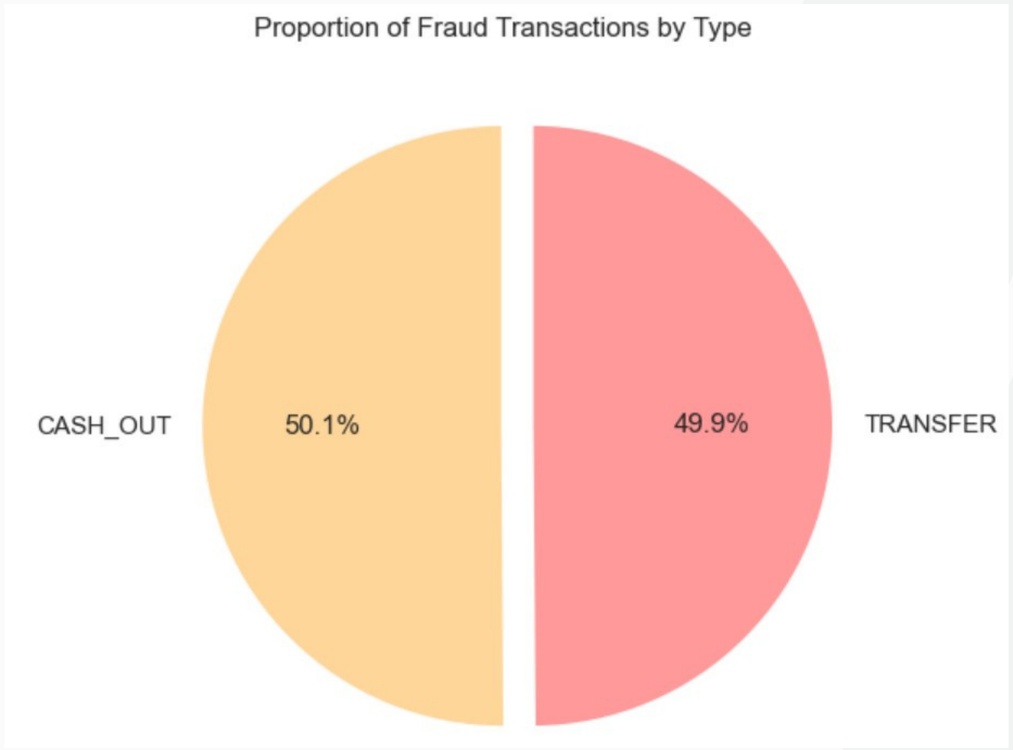
This highly imbalanced distribution, where the number of non-fraudulent transactions substantially outweighs the number of fraudulent transactions, can pose challenges during model training. It is crucial to address this class imbalance to ensure that the model does not become biased towards the majority class, leading to suboptimal performance in identifying the minority class (fraudulent transactions)

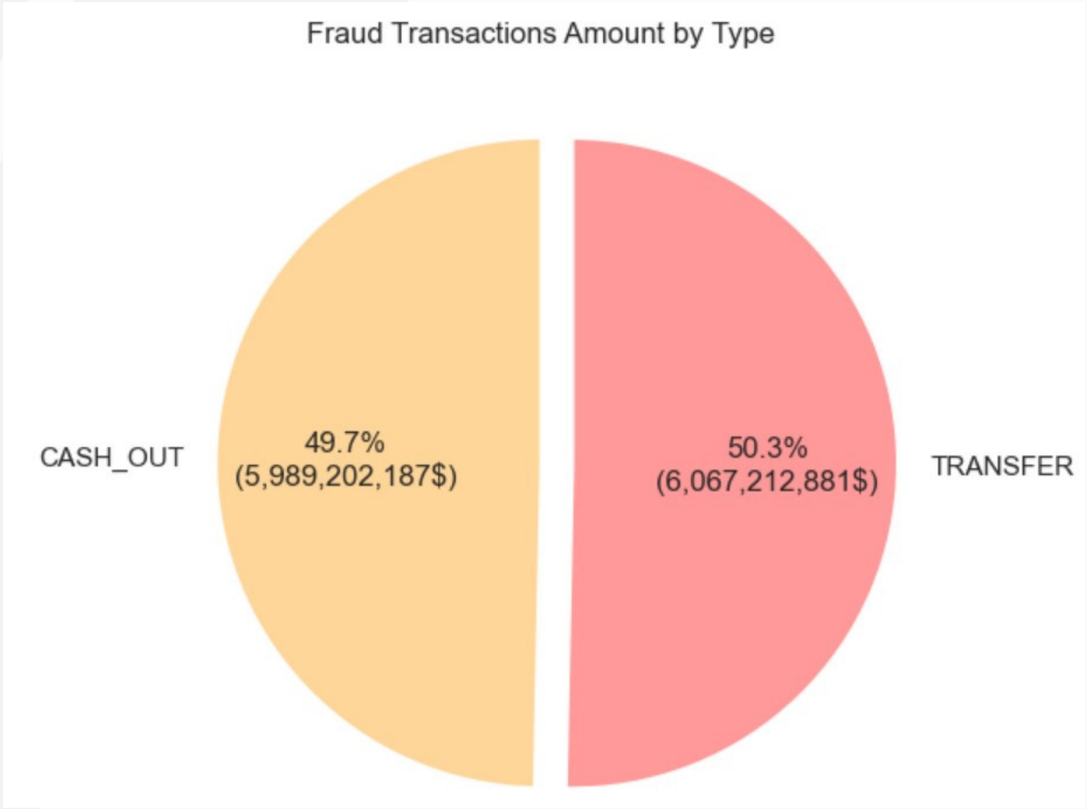
Fraud Transactions percentage: 0.13%
Non-fraud Transactions percentage: 99.87%



Exploratory Data Analysis

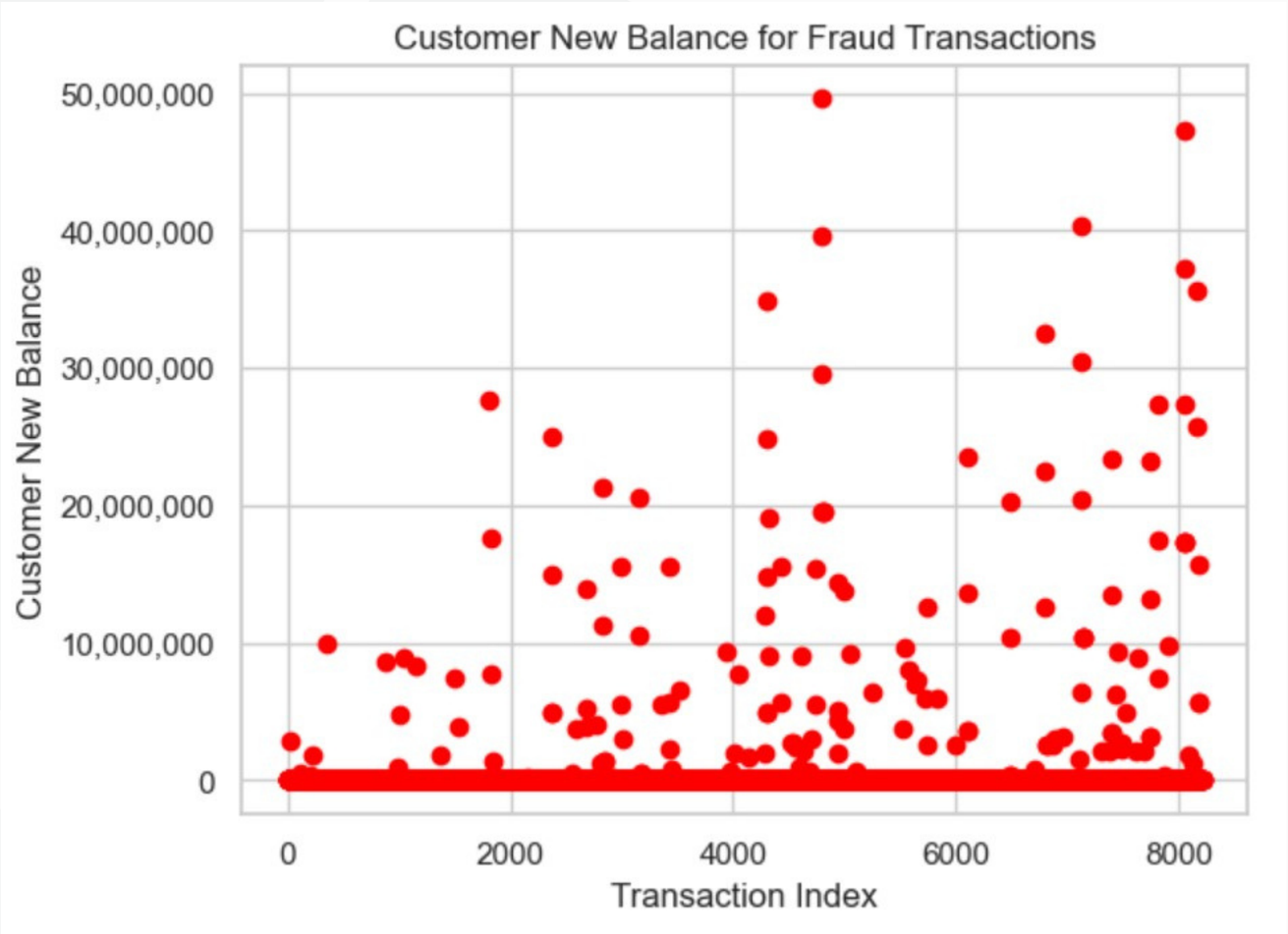
This concentration of fraud within CASH_OUT and TRANSFER transactions suggests that these specific types may be more susceptible to fraudulent activities





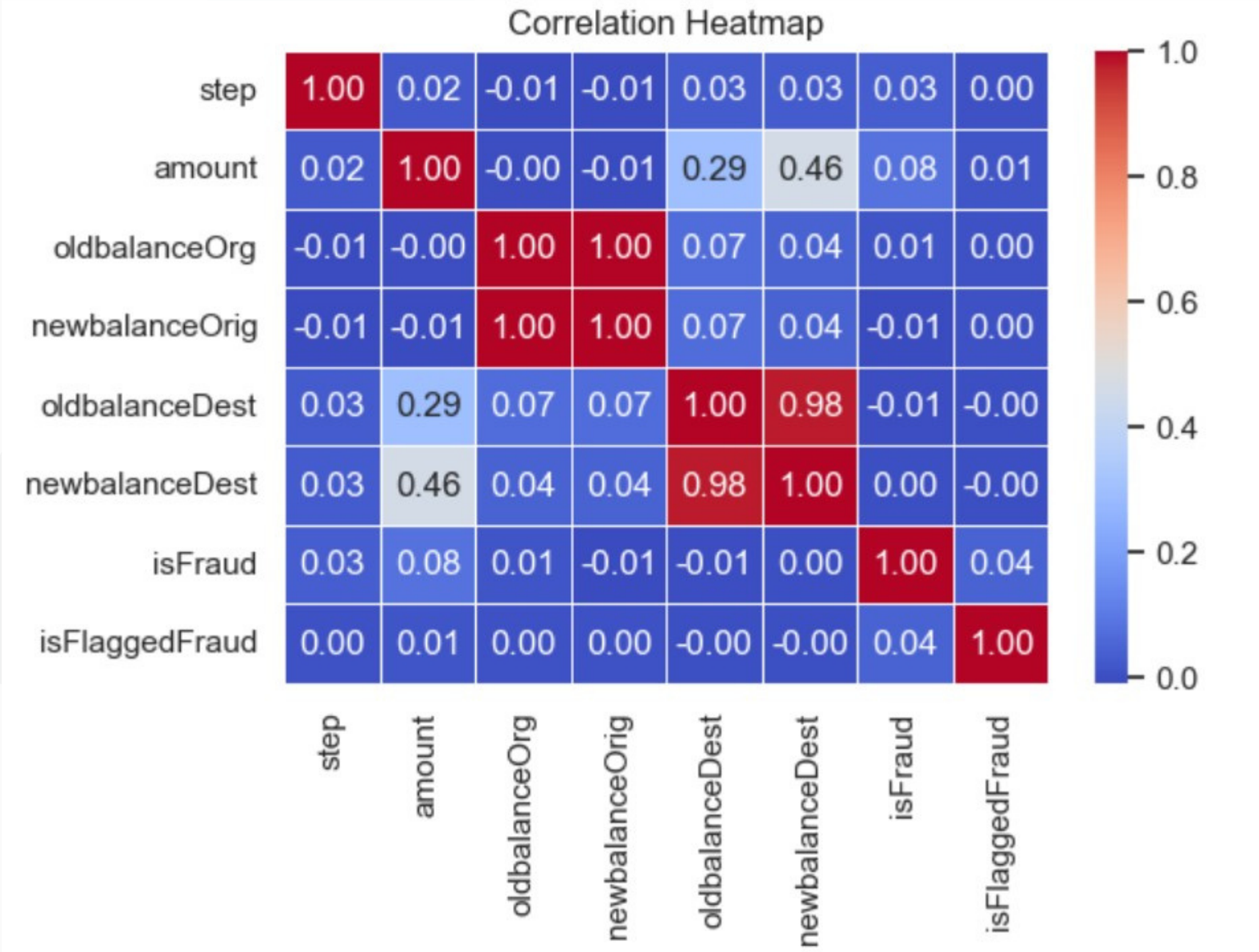
Exploratory Data Analysis

1. All fraudulent transactions fall into the category of very low amounts.
2. Many fraud transactions are linked to customers whose account balances drop to zero afterward. This pattern implies that fraudsters often manipulate transactions to deplete or minimize the affected customers' balances..



Exploratory Data Analysis

- oldbalanceOrg and newbalanceOrig has strong positive relationship.
- oldbalanceDest and newbalanceDest has strong positive relationship.
- oldbalanceOrg and amount has weak positive relationship.
- newbalanceOrig and amount has moderate positive relationship.



1 Data processing

We've substantially reduced the dataset, now containing 2,770,409 entries
converting categorical of the 'type' column data into a format suitable for machine learning models

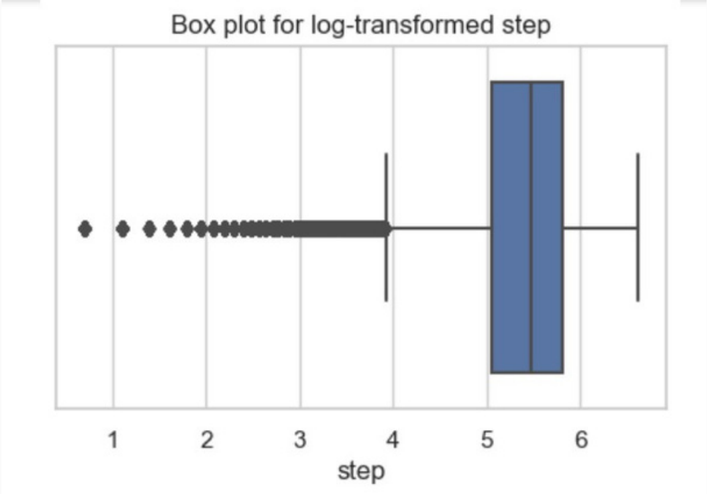
Data processing

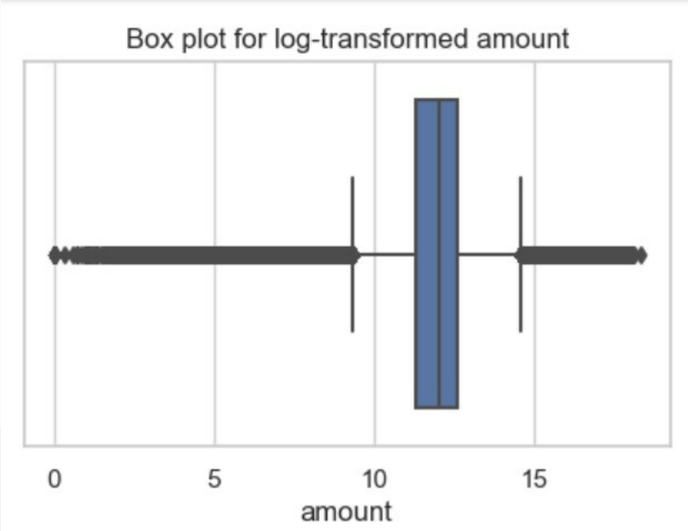
the distribution of log-transformed values for each column.

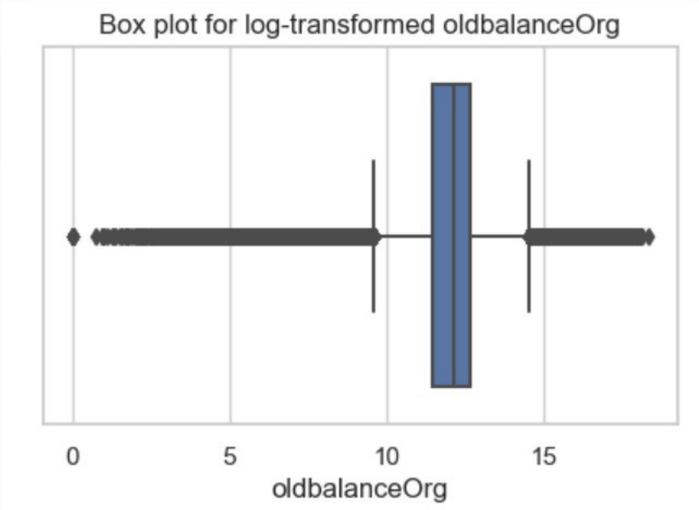
- The 'step,' 'amount,' 'oldbalanceOrg,' 'newbalanceOrg,' 'oldbalanceDest,' and 'newbalanceDest' columns were log-transformed using np.log1p() to make the distributions more symmetric.
- The presence of numerous points outside the box in the box plots indicates the potential existence of outliers or extreme values. These outliers may significantly deviate from the majority of the data and could impact the statistical analysis or modeling processes.

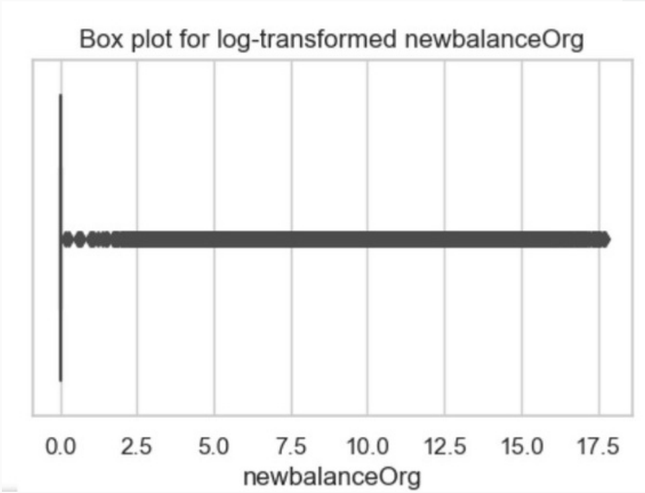
Data processing

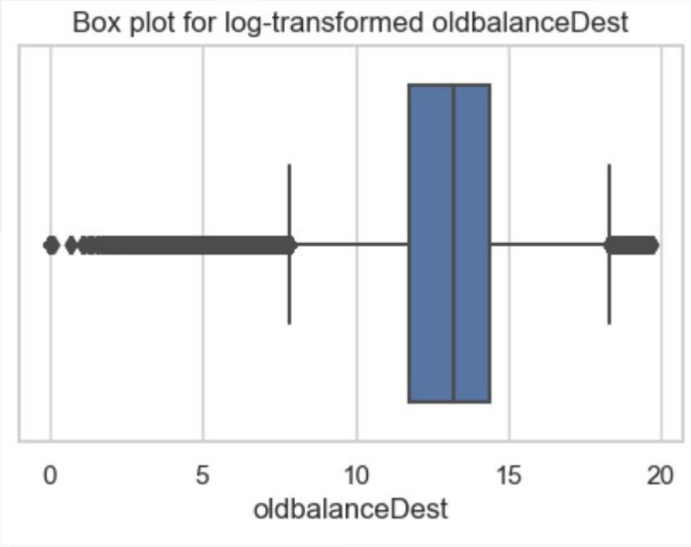
the distribution of log-transformed values for each column.

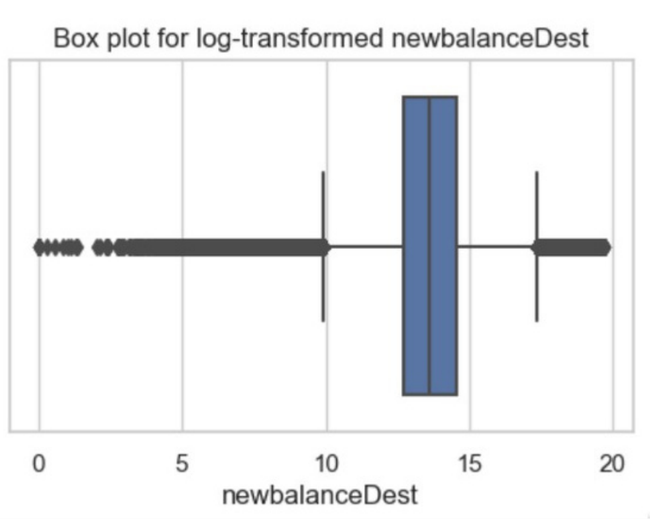












Data processing

the log1p transformation is applied to the data, addressing cases where the IQR is 0, thus enabling a more robust scaling mechanism. The combination of log1p and RobustScaler contributes to the preprocessing steps, enhancing the distribution of data and providing a resilient scaling approach suitable for features with potential outliers.

la plage interquartile (IQR)

IQR for step: 177.0
IQR for amount: 223817.64
IQR for oldbalanceOrg: 231324.15999999997
IQR for newbalanceOrg: 0.0
IQR for oldbalanceDest: 1607826.7
IQR for newbalanceDest: 1765652.54

3 Evaluating different models

Evaluating different models

	step	type	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	newbalanceOrg
2	-6.296978	1	-5.236120	-5.584706	0.0	-5.074675	-4.532829	1	0.0
3	-6.296978	0	-5.236120	-5.584706	0.0	-1.253342	-1.963333	1	0.0
15	-6.296978	0	0.222628	0.172787	0.0	-1.800822	-0.672214	0	0.0
19	-6.296978	1	0.175042	0.122596	0.0	-1.231467	-0.664175	0	0.0
24	-6.296978	1	0.457929	0.420966	0.0	-1.720506	-0.507408	0	0.0
...	...	...	...	...	...	...	...	...	...
6362615	1.508639	0	0.523707	0.490344	0.0	-5.074675	-0.471763	1	0.0
6362616	1.508639	1	2.758350	2.847284	0.0	-5.074675	1.103810	1	0.0
6362617	1.508639	0	2.758350	2.847284	0.0	-0.803156	1.109630	1	0.0

with imbalanced data, StratifiedKFold is advantageous for cross-validation. It randomly divides data into k folds while maintaining class proportions, promoting model training on a representative sample.

Evaluating different models

Handling Imbalance

To address this imbalance and enhance the efficiency of model training, random undersampling was performed. With a sampling strategy set at 0.1 (10% of the majority class), the new class distribution achieved a more balanced representation.

```
Class distribution before random undersampling:
0    2209757
1       6570
Name: isFraud, dtype: int64

Class distribution after random undersampling:
0    65700
1     6570
Name: isFraud, dtype: int64
```

using SMOTE here is to create additional synthetic examples of the minority class (fraud transactions) in the training data. This helps prevent the model from being biased towards the majority class.

```
0    2209757
1     220975
Name: isFraud, dtype: int64
```

Logistic Regression

Hyperparameter Tunning

- The grid search will optimize the hyperparameters based on both the F1 score, ROC AUC score and Recall.
- The final fitted model will be the one that achieved the highest F1 score during the cross-validation process.

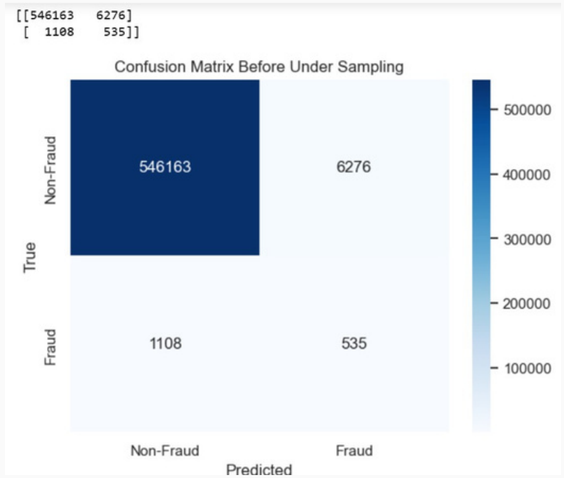
```
Before under sampling:
   Best Hyperparameters: {'C': 10, 'penalty': 'l2'}
   Best Score: 0.4433201731547515
After under sampling:
   Best Hyperparameters: {'C': 10, 'penalty': 'l2'}
   Best Score: 0.6526027397260273
```

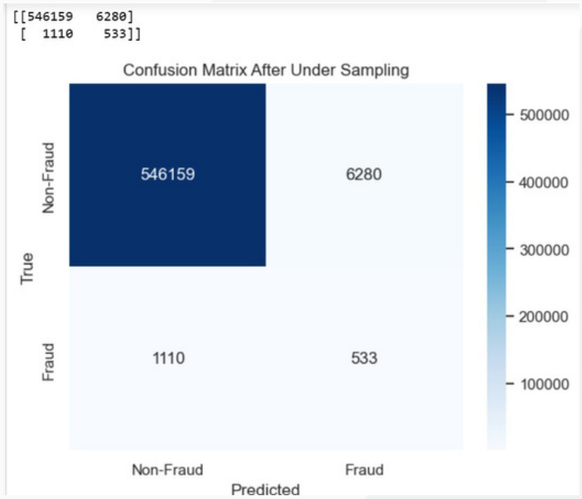
The performance evaluation of the Logistic Regression models on both oversampled and undersampled datasets reveals similar results.

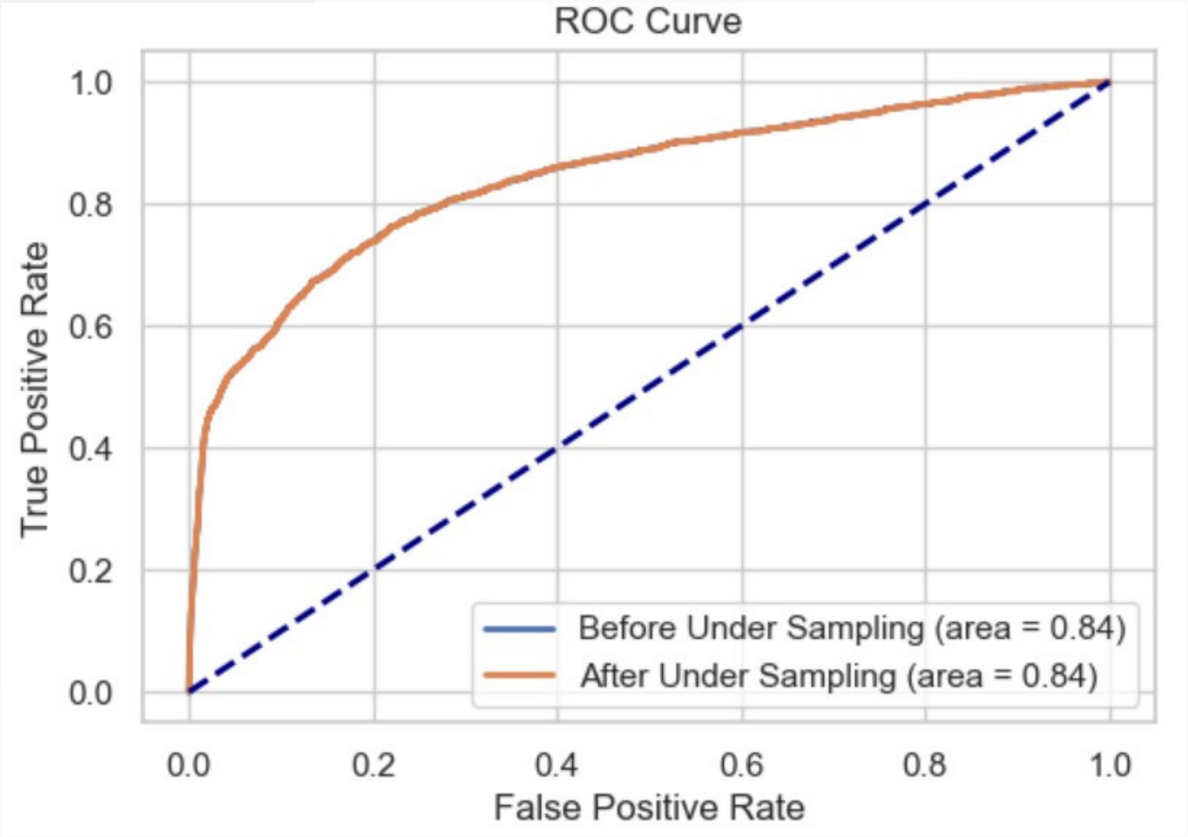
```
Oversampled:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99    552439
           1       0.08      0.33      0.13      1643

    accuracy                           0.99    554082
   macro avg       0.54      0.66      0.56    554082
weighted avg       1.00      0.99      0.99    554082

Matthews Correlation Coefficient: 0.16
ROC-AUC Score: 0.84

Undersampled:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99    552439
           1       0.08      0.32      0.13      1643

    accuracy                           0.99    554082
   macro avg       0.54      0.66      0.56    554082
weighted avg       1.00      0.99      0.99    554082

Matthews Correlation Coefficient: 0.15
ROC-AUC Score: 0.84
```

Logistic Regression

Logistic Regression can be used for imbalanced datasets, but its performance may be affected by the class imbalance. I
doesn't inherently handle class imbalance, and the model might be biased towards predicting the majority class.
We will try to use different model which can handle this very imbalanced data.







Random Forest

For highly unbalanced datasets, an ensemble method like Random Forest can often perform well. Random Forest is an ensemble of decision trees, and its ability to combine multiple trees can help mitigate the impact of class imbalance.

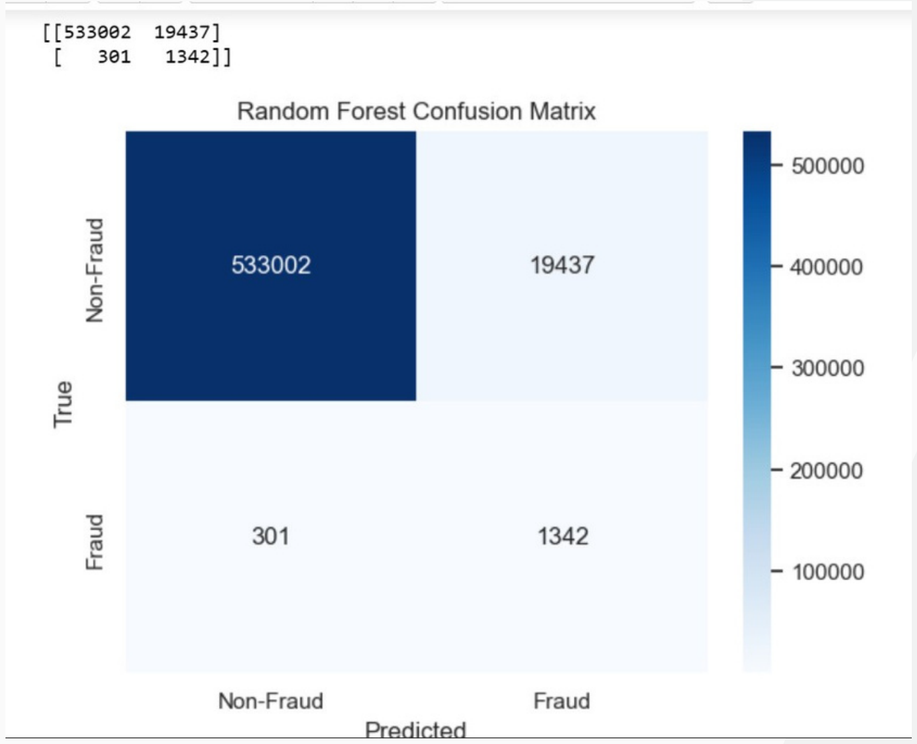
Hyperparameter Tunning

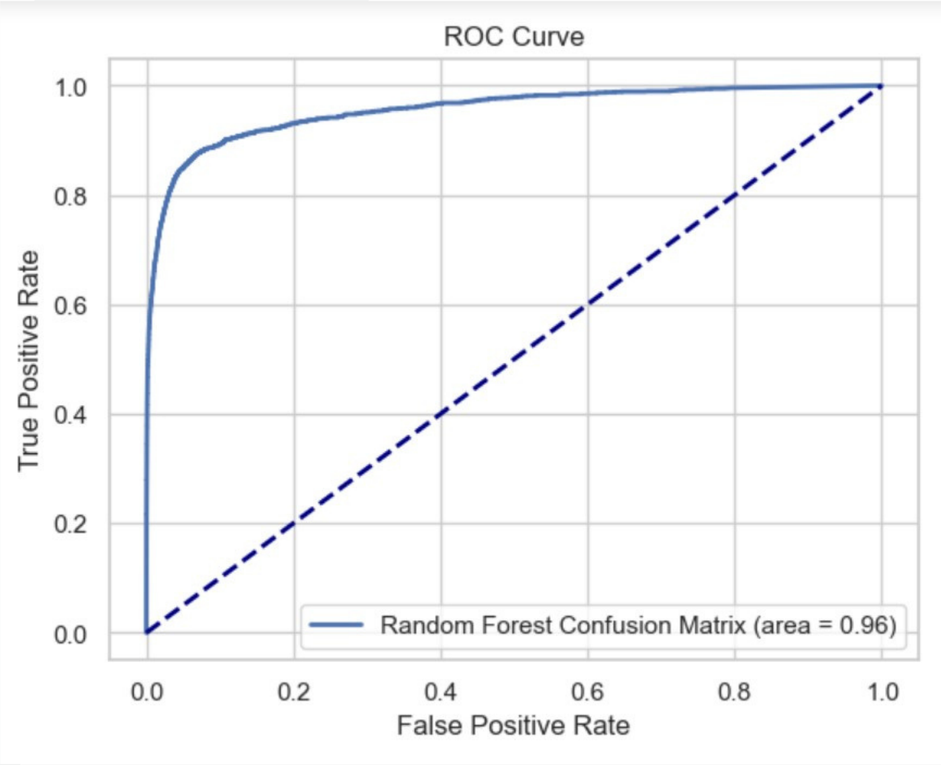
```
RandomForestClassifier(class_weight='balanced', min_samples_leaf=4,
                       min_samples_split=15)
```

```
              precision    recall  f1-score   support

           0       1.00      0.96      0.98    552439
           1       0.06      0.82      0.12      1643

    accuracy                           0.96    554082
   macro avg       0.53      0.89      0.55    554082
weighted avg       1.00      0.96      0.98    554082

Matthews Correlation Coefficient: 0.22
ROC-AUC Score: 0.96
```

Random Forest

- Random Forest obtains the highest score of all using K-fold cross-validation.
- The best performing model is Random Forest for identifying fraudulent and non-fraudulent payments, as the AUC is 0.999, which is close to 1. This means it has a good separability measure, and the model has an 99.9% chance of being able to distinguish between positive and negative classes..





XGBoost Model

Core Principles of XGBoost:

- Regularization: XGBoost incorporates regularization techniques to prevent overfitting, a common issue in machine learning models. This helps ensure that the model generalizes well to unseen data.
- Scalability: XGBoost is highly scalable, capable of handling large datasets and complex problems efficiently. It can be parallelized across multiple machines, making it suitable for large-scale machine learning applications.

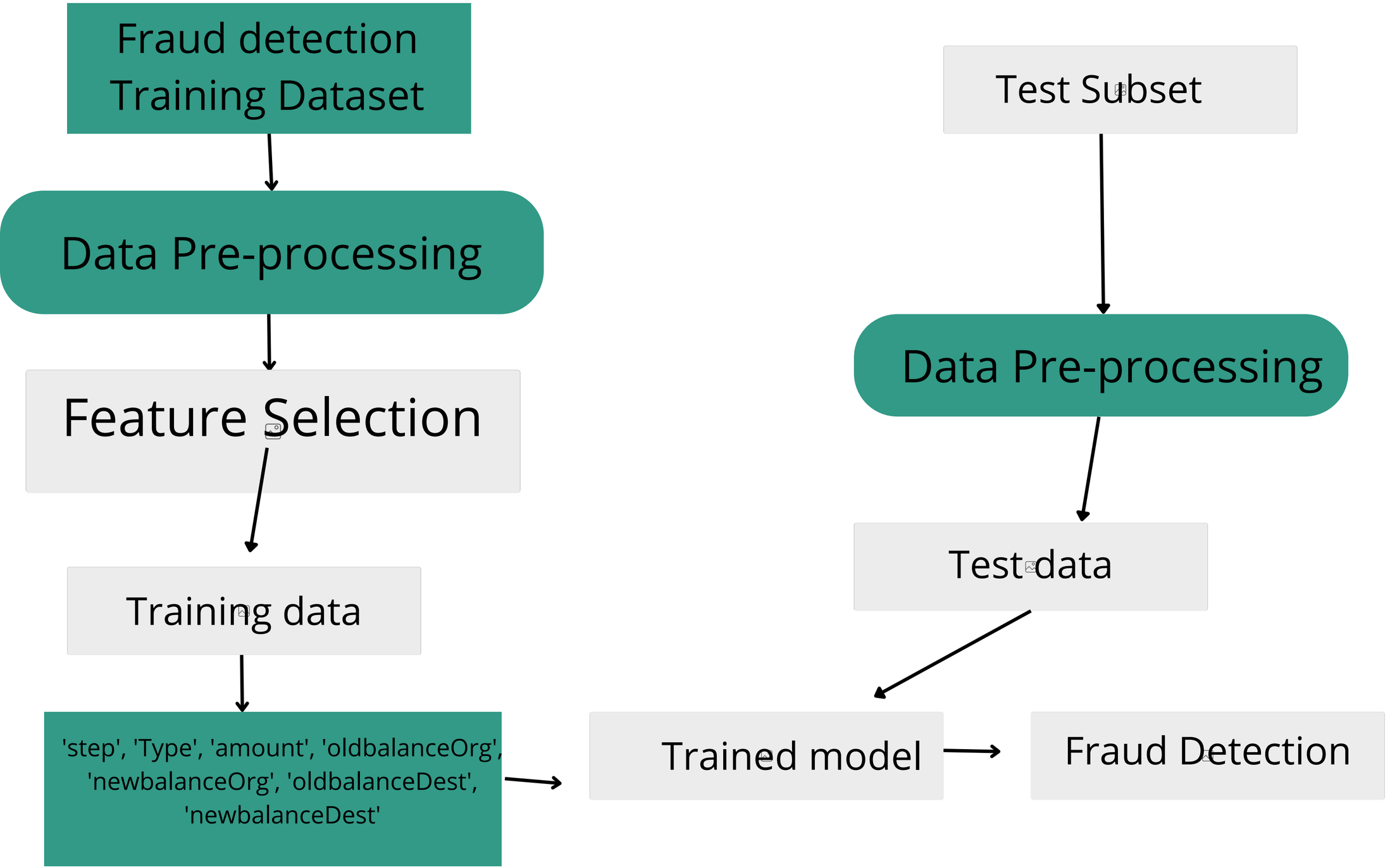
Hyperparameter Tunning

```
Best Hyperparameters: {'subsample': 0.9, 'scale_pos_weight':
5, 'n_estimators': 350, 'max_depth': 4, 'learning_rate': 0.3,
'lambda': 1, 'gamma': 5, 'colsample_bytree': 0.9, 'alpha': 0.
1}
```

the XGBoost and Random Forest models

The XGBoost model exhibits a notable improvement over the RandomForest model in several key performance metrics. Here's a comparative analysis.

- XGBoost achieves a higher True Positive count (1179) compared to the Random Forest model.
- XGBoost has a lower False Positive count (464) than the Random Forest model, indicating a better ability to correctly identify fraudulent transactions.
- Random Forest Model shows a lower True Positive count (1256) compared to XGBoost.
- Random Forest Model has a higher False Positive count (4779) than XGBoost, suggesting a greater tendency to classify non-fraudulent transactions as fraudulent.


Fraud detection
Training Dataset
Data Pre-processing
Feature Selection
Training data
'step', 'Type', 'amount', 'oldbalanceOrg',
'newbalanceOrg', 'oldbalanceDest',
'newbalanceDest'
Trained model
Test Subset
Data Pre-processing
Test data
Fraud Detection

thank you for your attention

Random Forest Model:

Advantages:

- High Precision for Non-Fraud Cases: The RandomForest model achieves high precision (100%) for non-fraudulent transactions, minimizing the rate of false positives in this category.
- Moderate Recall for Fraud Cases: While the recall for fraud cases is 76%, indicating its ability to capture positive instances, it is relatively lower than the XGBoost model.
- High Overall Accuracy: The model maintains high overall accuracy at 98%, showcasing its effectiveness in correctly classifying both classes.
- Decent MCC and ROC-AUC Score: The Matthews Correlation Coefficient (MCC) is 0.27, indicating a moderate overall performance. The ROC-AUC score is 95%, signifying a good ability to discriminate between classes.